

Borg Backup

Christian Neukirchen, Tristan Helmich

DevOps Meetup Munich · May 16, 2017



Why backups?

- Nobody wants *backup*, everyone wants *restore*.
- Who is running backups of their personal systems?
- Typical pain points:
 - Off-site storage
 - Storage overhead
 - Privacy & Confidentiality
 - Maintenance cost
 - Vendor lock-in

Why use Borg Backup?

Off-site storage

- SSH transport (needs `borg` on remote)

Storage overhead

- Compression
- Content deduplication
- Deleting archives is possible

Privacy & Confidentiality

- Client-side encryption
- Encrypted transport

Why use Borg Backup?

Maintenance cost

- Configurable pruning
- Non-interactive use possible
- Partial backups can be resumed

Vendor lock-in

- Open Source (3-clause-BSD)
- Python 3 (portable: Linux, *BSD, macOS)

How to use Borg Backup?

```
borg init user@hostname:backup
```

```
borg create user@hostname:backup::work-{now:%Y-%m-%d}  
/home/me --exclude '*~' --compression zlib
```

```
rm -rf /home/me/work # oops
```

```
borg list user@hostname:backup
```

```
cd /home/me
```

```
borg extract user@hostname:backup::work-2017-05-16 work
```

More features

- FUSE file system to browse contents of whole repository
- Time-based pruning (how many archives to keep per hour, day, week, month, year)
- Good metadata support: {a,c,m}times with nanosecond precision, xattrs (incl. caps and ACL), device nodes
- Backup/restore of streams via stdin/stdout
- Append-only repos
- Meta-data and data checking is possible (cryptographic signatures)
- Client-side encryption: passphrase (default), keyfile, none
 - also of meta data (file names, sizes)

Content deduplication

- Renaming and copying file is essentially free.
- Not only file-based deduplication, but content-based chunking.
- Detects insertions/deletion of *parts* of files.
- Great for:
 - append-only files, logs
 - VM images
 - mbox (even with deletions)
- Multiple hosts can backup into same archive
- Chunking is very customizable (if needed)

Content deduplication results

57 backups of a full notebook since 2014:

	Original size	Compressed size	Deduplicated size
This archive:	143.27 GB	115.84 GB	1.34 GB
All archives:	6.87 TB	5.66 TB	169.10 GB

384 backups of a development workstation since Mar 2016:

	Original size	Compressed size	Deduplicated size
This archive:	162.25 GB	112.37 GB	246.65 MB
All archives:	45.87 TB	29.16 TB	346.78 GB

	Unique chunks	Total chunks
Chunk index:	9359546	903415142

Downsides

- Needs free local disk space to create a backup (and for all other writing operations)
- No support for dumb remote storage, needs SSH and Borg installed
- Locking: no restore when a backup is running
- Sub-optimal for long-time-archiving (complex data format, needs `borg` to restore)

Alternatives

- Attic (predecessor to Borg, unmaintained, has critical bugs)
- Restic (very similar to Borg, in Go, beta)
- Duplicati (many dumb backends, in C#)
- rdedup (in Rust, public-key, no remote)

Questions?



<https://borgbackup.readthedocs.io/>

Thank you.